

# Razvoj linijskog dizajna u PostScriptu

---

Vagner, Tara

Undergraduate thesis / Završni rad

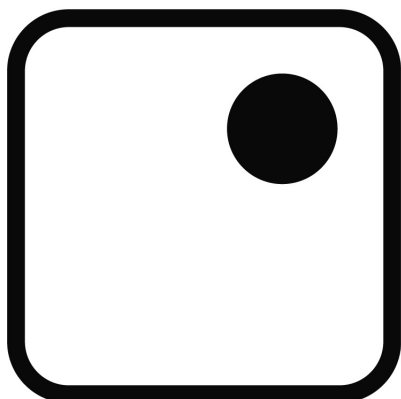
2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:905365>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-18**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



**SVEUČILIŠTE U ZAGREBU  
GRAFIČKI FAKULTET ZAGREB**

# **ZAVRŠNI RAD**

Tara Vagner

**SVEUČILIŠTE U ZAGREBU GRAFIČKI FAKULTET  
ZAGREB**

**Smjer: Tehničko-tehnološki**

**ZAVRŠNI RAD**  
**RAZVOJ LINIJSKOG DIZAJNA U POSTSCRIPTU**

**Mentor:**

Prof.dr.sc., Klaudio Pap

**Student:**

Tara Vagner

Zagreb, 2021.

## Sažetak

Ovim radom se prikazuje razvoj linijskog dizajna u PostScript programskom jeziku. Korištenjem linija i njenih atributa u različitim oblicima prikazuju se mogućnosti linijskog grafičkog dizajna te se je usporedila jednostavnost i brzina dizajniranja u PostScript jeziku, naprema dizajniranju u Adobe Illustratoru. U eksperimentalnom dijeu prikazuje se apstraktni dizajn napravljen pomoću linija. U radu se je pristupilo razvoju linijskog dizajna od najjednostavnijih do složenijih oblika, poput krugova, Bezierovih krivulja i ostalih. Eksperimentiralo se s različitim atributima linije kao što je debljina linije, sivoća, boja, crtkanost i linijski završetci. Istražila se mogućnost različitog, jednostavnog i veoma brzog dizajniranja pomoću samih linija.

## Sadržaj

1. UVOD .....	5
2. TEORIJSKI DIO.....	6
2.1. PostScript .....	6
2.2. Adobe Illustrator.....	8
2.3. Vektori.....	9
3. PRAKTIČNI DIO .....	11
3.1. Osnovne naredbe .....	11
3.2. Postavljanje radne površine .....	12
3.3. Bezierove krivulje.....	14
3.4. Memorijski mehanizmi -gsave i grestore.....	17
3.5. Atributi za linije.....	17
3.6. Kružnice i translacije .....	20
3.7. Korišćenje repeat petlje .....	22
3.8. Korišćenje for petlja .....	28
3.9. Učahureni postscript - eps .....	30
4. REZULTATI I RASPRAVA.....	31
5. ZAKLJUČAK .....	33
6. LITERATURA .....	34
7. PRILOZI.....	35

## 1. UVOD

Cilj ovog završnog rada je prikaz razvoja linijskog dizajna u PostScriptu jeziku, te usporedba jednostavnosti dizajniranja naprema Adobe Illustratoru. Prikazuju se sve vrste linija od onih najjednostavnijih do složenijih oblika. Korištenjem linija i njenih atributa prikazala sam mogućnost različitog grafičkog dizajna kroz naredbe. Eksperimentalni dio prikazuje ravne linije s različitim atributima ( debljina linije, sivoća, boja, crtkanost i linijski završetci ), krugove, Bezierove krivulje, te različite petlje. Sve te linije zajednički čine apstraktivni dizajn.

Također se prikazuje vremenska razlika potrebna za dizajniranje istog oblika kroz programske naredbe u PostScriptu i kroz crtanje u Adobe Illustratoru. Uočava se razlika u dizajniranju, te prikazuje zanimljivost Adobe Illustratora i PostScript jezika.

U poglavlju osnovne naredbe se dobiva uvod u PostScript jezik, prikazuju se njegove najosnovnije i najpotrebitije naredbe, bez kojih je ne moguće bilo što drugo napraviti. Postavljanje radne površine prikazuje i objašnjava naredbe kroz koje je izrađena radna površina, te kako je ona zadana u Adobe Illustratoru. U poglavlju Bezierove krivulje se objašnjava i prikazuje dizajniranje uvinutih linija. Kroz memorijske mehanizme se objašnjava punjenje i čišćenje stacka, te čemu to služi. Atributi za linije prikazuju sve mogućnosti uređivanja i mijenjanja linija na različite načine. U poglavlju kružnice i translacije se prikazuje mogućnost dizajniranja kruga, kružnica i kružnih isječaka, te kroištenja translacije. Korištenje repeat petlje prikazati će razne načine dizajniranja ponavljanjem istog objekta na različite načine. U poglavlju korištenje for petlje će se također pokazati ponavljanje jednog objekta, no u for petlji je taj objekt promijenjiv, pa mu se kroz petlju mijenjaju razni atributi. Učahureni PostScript nam prikazuje korištenje dizajna napisanog kroz naredbe u vektorskom obliku.

## 2. TEORIJSKI DIO

### 2.1. PostScript

Računalna grafika definirana je kao vid grafike u kojoj se za generiranje, memoriranje, obradu i prezentiranje slikovnih sadržaja koristi računalo. Iako se računalna grafika u pravom smislu riječi pojavljuje 60-ih godina prošloga stoljeća, kada nastaju prvi računalno animirani film i prva video igra, neki njezine začetke pronalaze već u 19. stoljeću, kada je James Joseph Sylvester osmislio matičnu notaciju, često korištenu u algoritmima računalne grafike. [1] PostScript je grafički programski jezik kojega Adobe Systems predstavlja tek 1984. godine kao svoj prvi komercijalni proizvod. Tada PostScript postaje industrijski standard za izradu slika i tisak. Svi veći proizvođači pisača proizvode pisače koji sadrže ili se mogu učitati sa programom PostScript, koji također radi na svim većim platformama operacijskih sustava. Datoteka Postscript može se prepoznati po njenom nastavku ".ps". Postscript opisuje tekst i grafičke elemente na stranici na crno-bijelom ili pisaču u boji ili drugom izlaznom uređaju, poput snimača dijapozitiva, alata za postavljanje slika ili zaslona.



Slika 1. PostScript pisač

( Izvor: <https://blog.inkjetwholesale.com.au/prINTER-education/what-is-postscript-printing/> )

Postscript obrađuje industrijski standardni, skalabilni font u formatima Type 1 i TrueType. Korisnici mogu pretvoriti datoteke Postscript u Adobe Portable Document Format (PDF) pomoću Adobe Acrobat. PDF datoteke prikazuju ispis dokumenta na zaslonu. [2]

Ovaj grafički programski jezik nudi mogućnost isertavanja vektorskih oblika korištenjem jednostavnih naredbi. PostScript je programski jezik namjenjen oblikovanju grafike na stranicama za tisak, no ima sličnu strukturu komandi kao Basic, Pascal i C. Ima svoj specifičan grafički jezik koji je nastao razvojem računarske tipografije, fotosloga i računarske reprofotografije. U PostScriptu kako bi mogli odrediti grafički oblik slova ili slike koristi se komandni jezik koji se sastoji od puta, parametara i komandi koji ga određuju. Komande su pisane kurentnim i verzalnim slovima na engleskom jeziku ili kraticama izvedenih iz engleskih riječi. Također, neke od komandi se sastoje od više engleskih riječi ili kratica ( npr. `sethsbcolor` ). Parametri su numeričke vrijednosti koje se pišu ispred komandi, te oni određuju: gdje komanda počinje, završava, koliko traje, kako se prikazuje i kako djeluje. Razmak između parametra i komandi se mora poštovati, jer inače naredbe neće funkcionirati. Niz komandi može se kontinuirano pisati u istom retku, no zbog preglednosti i lakšeg snalaženja puno je bolje odvajati na više dijelova. `%` služi kao komentar, koji isto tako pomaže programeru kao podsjetnik ili za olakšavanje snalaženja u programu. Komentar prestaje biti aktivan prelaženjem u novi redak, te ne poremećuje niti jedan dio naredbi.

Koordinatni sustav koji koristimo u PostScriptu kreće iz donjeg lijevog ugla, te je zasnovan na točkama sa  $x$  i  $y$  koordinatama. Standardna veličina točke određena je preko inča. Inč ima 72 točke, točnije on je podijeljen u 6 dijelova nazvanih pica, a ukupno iznosi 12 pica. Dakle jedna točka iznosi 0.353 milimetra.

PostScript jezik može se pisati u različitim tekst editorima koji mogu zapisati tekst u čistoj ASCII formi, odnosno bez ikakvih tipografskih zahvata, a pregledavamo ih u Ghostscriptu ili GSview. [3]

Ono što PostScript čini naprednim u odnosu na programe koji koriste grafičko korisničko sučelje pri definiranju oblika je mogućnost da se željeni elementi programski odrede s promjenama kroz petlju i to uz savršenu preciznost. Svakom je elementu moguće promijeniti parametre poput boje, veličine ili debljine linije kroz određeni broj ponavljanja. Takve mogućnosti svoju primjenu i danas nalaze u zaštitnom tisku. Jedan od načina zaštite korištenjem PostScripta je generiranje oblika čije linije mogu sadržavati jedinstveni isprekidani uzorak. Intervali razmaka između isprekidanih linija koje čine putanju može se izvaditi iz polja, a može činiti i nasumične brojeve. Takav je uzorak vrlo teško ponoviti. PostScriptu ne nedostaje mogućnosti u odabiru boja, te je boju moguće



dobiti u svakoj željenoj vrijednosti. PostScript podržava RGB, CMYK i HSB sustave boja, a svakom je elementu uz boju moguće promijeniti i opacitet. [1].

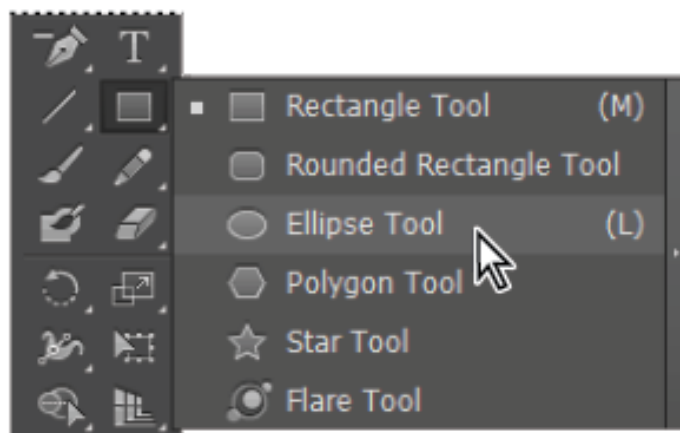
## **2.2. Adobe Illustrator**

Kada je 1984. godine Adobe predstavio PostScript, koji je kao grafički alat odgovarao potrebama toga vremena, istodobno započinje razvoj Adobe Illustratora, koji će biti predstavljen 1987. godine kao program namijenjen izradi logotipa i grafičkom dizajnu. Prva je verzija izdana pod kodnim imenom „Picasso“, a već sljedeće godine izlazi nova verzija s velikim brojem novih alata. Adobe već od 1993. nadogradnjom unosi mogućnost razvrstavanja elemenata u slojevima, 3 godine kasnije uvodi mogućnost odabira boje s ekrana, ispune boje i korištenje gradijenta, a tek se 2008. godine pojavljuje mogućnost stvaranja većeg broja površina za crtanje. Brojni su osnovni alati uvedeni već u prvoj verziji, a kasnije nadogradnje većinom donose inovacije koje popravljaju ranije nastale probleme, olakšavaju korištenje postojećih i ubrzavaju rad. Adobe Illustrator je vektorski program namijenjen za crtanje različitih grafičkih oblika. Ovaj je program poznat među grafičkim dizajnerima zbog kvalitete boja kod izlaznog proizvoda. Vektorizirane ili vektorske slike sastoje se od točaka u virtualnom prostoru kojima se pridružujemo putem, kako bismo ih kasnije ispunili i tako dobili visokokvalitetne slike koje su koherentne u bilo kojoj veličini. Illustrator može prikazati CMYK paramater u boji na zaslonu RGB monitora gotovo 100% precizno s ispisom. Dok su u drugim vektorskim aplikacijama, dizajneri morali učiniti puno upravljanja konfiguracijskim bojama monitora prije postizanja gotovo precizne boje u ispisu. [4]

Adobe Illustrator je prvobitno profesionalni program za dizajn i crtanje zasnovan na vektorima. Koristeći se kao dio većeg tijeka dizajna, Illustrator omogućuje stvaranje svega, od pojedinačnih elemenata dizajna do cijelih skladbi. Dizajneri koriste Illustrator za izradu plakata, simbola, logotipa, uzoraka, ikona itd. Iako je među najpoznatijim alatima koji se koriste u dizajnu upravo zbog niza mogućnosti koje omogućuju kvalitetan rad, veliki nedostatak mu je u visokoj cijeni, što ga čini nedostupnim početnicima koji bi ga koristili u svrhu učenja. [5]

Sidebar na lijevoj strani prozora Illustrator naziva se ploča s alatima, postoji mnogo alata za odabir, izradu i manipuliranje objektima u radnom području (Artboard). Za grafičkog dizajnera poznavanje odjeljka s alatima vrlo je važno osnovno tehničko znanje,

jer kad radi mora pamtit i naviknuti se na mjesto gdje se nalaze ikone alata koje je potrebno kako bismo mu mogli brzo pristupiti. Također kao što u sidebaru imamo alate, tako imamo i palete transparentnosti, paletu poteza, paletu gradijenta, paletu slojeva. [6]



Slika 2. Dio Illustrator side bar-a

( Izvor: <https://www.agitraining.com/adobe/illustrator/tutorials/combining-shapes-using-the-shape-builder-tool-in-illustrator> )

Sve u svemu Adobe Illustartor je vrhunski program za vektorsko crtanje. Vektorski program stvara oblike sastavljene od točaka, linija i krivulja koje su pohranjene kao matematičke upute, za razliku od programa za bitmape, koja stvara oblike sastavljene od točaka piksela. Širok raspon tiskara i dizajnera koristi Illustrator za stvaranje dizajna i umjetničkih djela. [7]

## 2.3. Vektori

U današnjem digitalnom svijetu postoje dvije vrste grafika, digitalna i vektorska. Vektorska grafika se temelji na principu geometrije iliti vektora. Svaki vektor ima svoju početnu točku, smjer i završnu točku, duljinu. Ako je vektor zakrivljen onda sadrži i točke koje definiraju krivulju ili kuteve. Vektorskom grafikom stvaramo geometrijske oblike krug, kvadrat, trokutu i svakakve različite oblike.

Kako se vektorska grafika ne bazira na rasteru možemo je u beskonačnost povećavati ili smanjivati bez da ona izgubi na kvaliteti. To je zato što se bazira na matematičkim funkcijama, tj. o apsolutnoj udaljenosti između točaka. Poznata je još i pod imenom Loss Less grafika (grafika bez gubitaka). Idealana je za izradu logotipova, jer logotipe umanjene stavljamo na vizitke ili uvećane na jumbo plakete. Vektorska grafika je

savršena za jednostavne ili složene crteže koji ne trebaju biti fotorealistični. Također vektorska grafika je lako promijenjiva bez gubitaka na kvaliteti. Kao što ju možemo povećavati i smanjivati, isto tako ju lako možemo promijeniti ili ispraviti. [5]

Rasterska grafika pri povećavanju gubi na kvaliteti, točnije, počinju se vidjeti sitne kockice koje nazivamo rasterima, što se kod vektorske grafike ne može dogoditi jer se temelji na matematičkim funkcijama.



Slika 3. Vektorska i rasterska grafika

( Izvor: <https://excentric-design.net/vektorski-logo/> )

### 3. PRAKTIČNI DIO

#### 3.1. Osnovne naredbe

Kako bi lakše objasnila izradu svoga praktičnog dijela, prvo bi objasnila par osnovnih naredba koje se koriste skoro u svakoj složenijoj, te bez njih nije moguće izraditi skoro niti jedan dio programa.

100 300 move to

100 ovdje prikazuje x koordinatu, a 300 y. Moveto je naredba koja određuje početnu točku, odnosno pozicioniranje.

100 300 move to

300 400 lineto

Lineto crta liniju, koja ima nevidljivu stazu.

100 300 move to

300 400 lineto

stroke

Stroke daje vidljivost staze. Ako ne odredimo debljinu i boju, program sam uzima veličinu od 1 tipografske točke i crnu boju.

10 setlinewidth

10 prikazuje veličinu od 10 tipografskih točaka, a setlinewidth je naredba za određivanje debljine linije.

Kada određujemo sivoću koristimo setgray. 1 setgray je bijelo, dok je 0 setgray crno. 0,7 setgray nije 70% sivo, nego 30% zato što radi po svjetlosti.

Closepath je naredba koja služi za zatvranje staze, zbog moveto postscript zna gdje smo ju započeli. Closepath puno pomaže kod složenijih crteža, kada se teško raspoznaje gdje je početak, a ovom slučaju sa closepath program ga sam pronalazi. Fill popunjava cijelu stazu bojom, te oduzme pola strokea. Iako ako prvo napravimo fill, pa stroke, stroke će ostati cijeli.

### 3.2. Postavljanje radne površine

Kao što već u teoretskom dijelu je objašnjeno PostScript se koristi točkama koje se određuju u koordinatnom sustavu, no kako bi uopće mogli izraditi neki grafički dizajn prvo moramo odrediti taj koordinatni sustav, točnije nacrtati mrežu.

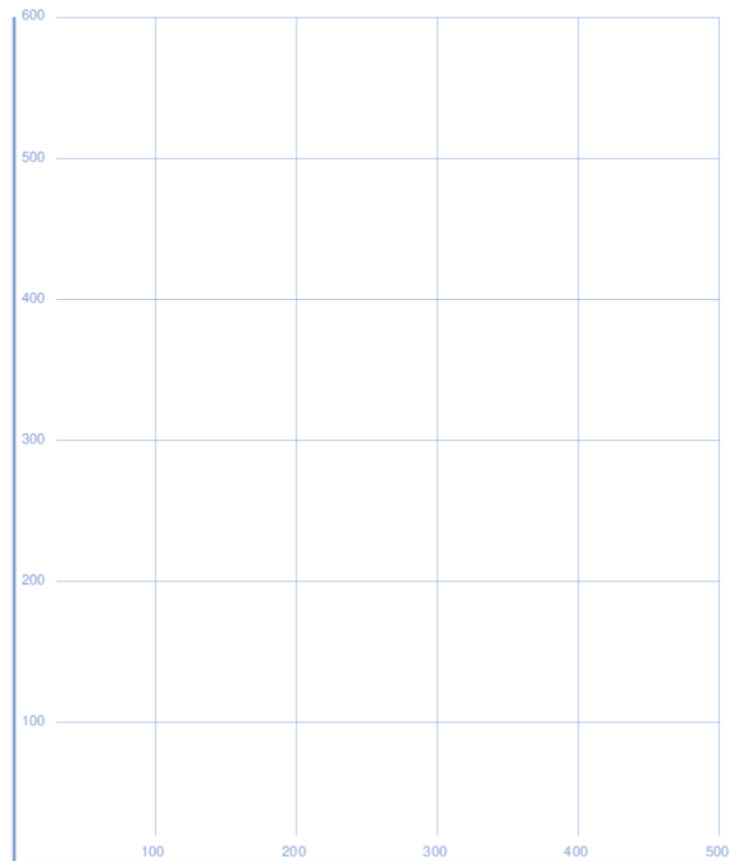
```
10 10 translate
/mreza {0.5 0.3 0 0 setcmykcolor
/xOs 500 def
/yOs 600 def
gsave
2 setlinewidth xOs 0 moveto 0 0 lineto 0 yOs lineto stroke
grestore
gsave
0.3 setlinewidth
6 { 30 100 moveto xOs 100 lineto stroke 0 100 translate } repeat
grestore
gsave
0.3 setlinewidth
5 { 100 20 moveto 100 yOs lineto stroke 100 0 translate } repeat
grestore
gsave
/tekst 3 string def /Helvetica findfont 10 scalefont setfont
100 100 yOs { /y exch def 5 y 2 sub moveto y tekst cvs show } for
```

```
90 100 xOs { /x exch def x 5 moveto x 10 add tekst cvs show } for
```

```
grestore
```

```
0 setgray } bind def
```

Mreža



Slika 4. Mreža u PostScriptu

( izvor: Tara Vagner )

Kao što vidimo ova mreža je 500x600 točaka, te je svijetlo plave boje.

Horizontalne linije počinju 30 točaka udaljene od nultočke. Svaki repeat petlje translacija koordinatni susstav vertikalno za 100 točaka, a vertikalne linije počinju na visini od 20 točaka.

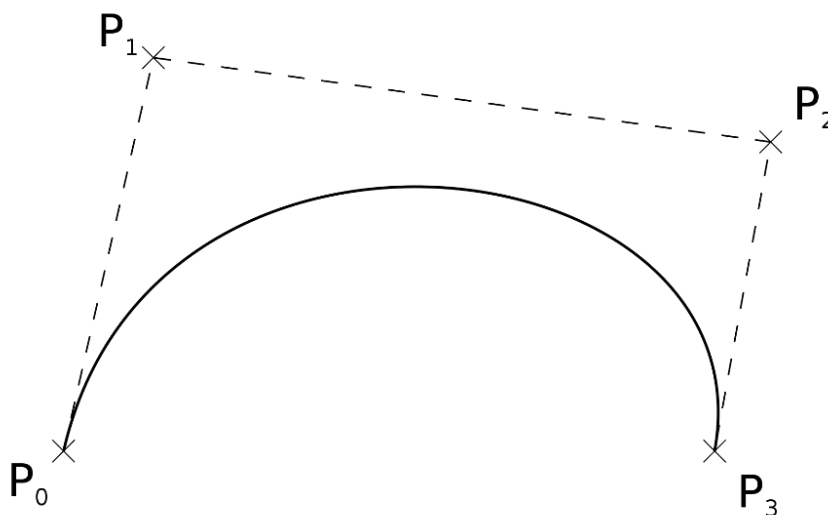
Ispisivanje brojka riješeno je for petljom gdje je sa „tekst 3 string“ određena duljina teksta od 3 znaka. Komanda cvs pretvra numeričku vrijednost u string. Ispisuje se u daljini od 5 točaka, a za 2 je pomaknut prema dolje ( 5 y 2 sub moveto ). Oblik fonta ( Helvetica ) i njegovu veličinu ( 10pt ) smo odredili kroz proceduru /Helvetica findfont 10 scalefont setfont. [3]

Također u Adobe Illustratoru sam kod odabira površine na kojoj ću raditi se odlučila za 500x600pt ( točaka ). Kod ovoga programa odmah pri otvaranju nove radne površine on prikazuje prozor preko kojeg se određuje kolika će nam biti radna površina i kako će biti orijentirana.

### 3.3. Bezierove krivulje

Početak i većina mog grafičkog dizajna se temelji na Bezierovim krivuljama. One u ovom slučaju određuju oblik glave i obrva. Unutar računalne grafike Bezierove krivulje su važan alat, kojim se služi većina računalnih programa pri oblikovanju jasnih glatkih krivulja, koje se mogu beskonačno skalirati. Kod bezierovih krivulja služimo se putanjama. [3]

Bezierova krivulja ima četiri točke, dvije fiksne i dvije natezne. Naredba za bezierovu krivulju je curveto.



Slika 5. Bezierova krivulja

(izvor: [https://hr.wikipedia.org/wiki/B%C3%A9zierova\\_krivulja](https://hr.wikipedia.org/wiki/B%C3%A9zierova_krivulja) )

%oblik lica

120 400 moveto

80 180 190 110 250 100 curveto

Stroke

380 400 moveto

420 180 310 110 250 100 curveto

stroke

Moveto je naredba koja nam određuje točku od koje krećemo, kod Bezierovih krivulja to je prva fiksna točka. Sljedeće dvije točke su natezne, one određuju u kojem će smjeru krivulja ići, a zadnja točka je ponovno fiksna i tamo bezierova krivulja završava.

Pošto je oblik lica trebao biti simetričan y parametri su isti kod obe krivulje, te su završne točke jednakih vrijednosti ( brada ). U glavi je vrlo lako predočiti gdje treba staviti natezne linije za oblik koji želimo dobiti, tako da dok sam samo gledala u koordinatni sustav lako sam odredila koje će to biti 4 točke. Ako se iz prvog pokušaja ne dobije linija kakva se želi ona se lako popravljja, jer tada vidimo prema kuda želimo da nam bude više zategnuta ili pomaknuta.

Oblik lica u Adobe Illustratoru radila sam pomoću pen toola s kojim je bilo vrlo jednostavno dobiti jednu stranu lica u željenom obliku. Pen tool je alat kojim postavljamo sidrišne točke i iscrtavamo ravne ili zakrivljene linije. Pomoću direct selection toola možemo pomicati bilo koju točku i zakriviti ju u smjeru kojem želimo. Na taj način lagano se popravljja putanja i oblik linije. Drugu stranu lica sam dobila tako da sam prvu kopirala i preko naredbe transform ( koju imamo desnim klikom miša na traku ), te onda reflect – vertical.

U oba programa se na vrlo jednostavne načine mogu dobiti zamišljene zakrivljene linije, te se one lako popravljaju i pomiću.

% vrat

200 120 moveto

200 80 200 30 150 0 curveto

stroke

300 120 moveto

300 80 300 30 350 0 curveto

Stroke



Vrat je 100 tipografski točki udaljen što možemo uočiti iz početne točke. On je također simetričan tako da je y svugdje isti, krivulja se širi po putanji, tako da kod druge natezne točke, kada uspoređujemo lijevu i desnu stranu, numerička vrijednost x se povećava za 100, a kod završne točke za 200.

%usi

112 300 moveto

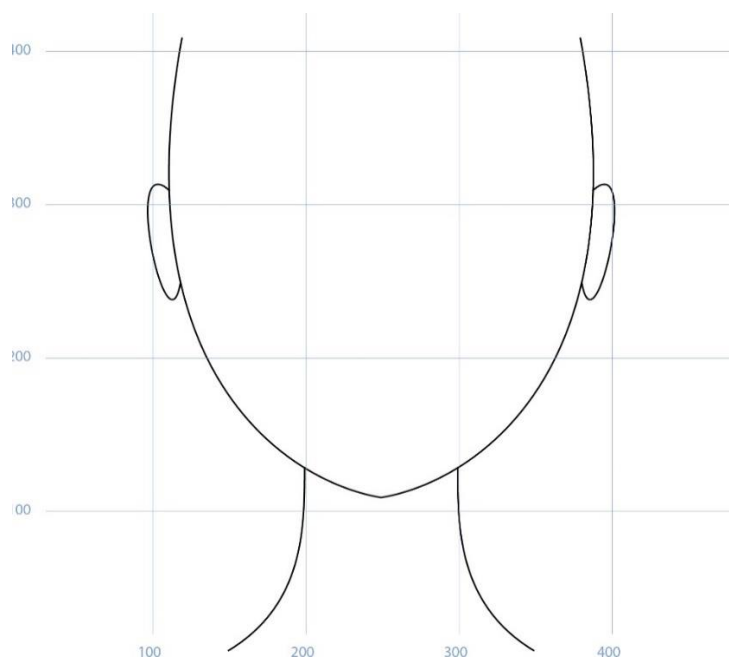
80 330 110 190 119 240 curveto

stroke

388 300 moveto

420 330 390 190 381 240 curveto

Stroke



Slika 6. Bezierova krivulja

( Izvor: Tara Vagner )

Isto kao vrat i lice, uši su simetrične, dakle numeričke vrijednosti y su jednake, a x se mijenja. Kod uši natezne linije su povučene više prema gore ili dolje kako bi mogli dobiti takav zaobljeni oblik.

Vrat i uši sam također kao i oblik lica u Adobe Illustratoru nacrtala pomoću pen toola. Kod uši je trebalo malo više točaka i popravljajanja zakrivljenosti, jer je malo više zaobljeniji i kompliciraniji oblik.

### **3.4. Memorijski mehanizmi - gsave i grestore**

Stack grafičkih stanja koriste samo operatori gsave i grestore kako bi održali stanje grafike. Operator gsave učinkovito gura (push) trenutno stanje grafike na stack grafičkih stanja, čineći ga kasnije dostupnim za grestore. Stack grafičkih stanja svojom stožastom prirodom omogućuje da se gsave i grestore mogu gnijezditi do limita stacka grafičkih stanja, što obično iznosi 20 razina ili više. U Display PostScript ekstenzijama jezika PostScript, uveden je koncept objekta grafičkog stanja, koji omogućuje da se trenutno grafičko stanje uhvati u objektu koji se može tretirati kao bilo koji drugi objekt. Postoje pridruženi operatori poput setgstate koji će uzeti objekt grafičkog stanja i postaviti sve trenutne parametre prema njegovom sadržaju. Ovo je puno fleksibilnije nego stack grafičkih stanja, jer omogućuje trenutačne promjene grafičkog stanja bez potrebe za push and pop operacijama. [8]

Ako želimo da program zapamti način na koji nešto crtamo i put koji radimo koristimo naredbu gsave prije svega ostalog, koja sprema tkuće grafičko stanje. Grestore koristimo na kraju tog dijela koda i onda on obnavlja zadnje spremljeno grafičko stanje. Tako možemo spremati i objavljujati grafička stanja po redosljedu spremanja, to se sve izvršava preko stacka. Sa grestore na taj način cijelo vrijeme čistimo stack. Komanda gsave sprema sve parametre na vrh stacka, a grestore ih sve skida. [3]

U praksi kada bi željeli da nam određeni dio koda bude zelene boje, a drugi plave, bez korištenja gsave i grestore, na vrhu stacka će ostati npr. zelena i sve će nam biti zelene boje i izgledati će kao da plavu nismo ni primjenili.

### **3.5. Atributi za linije**

%obrv

gsave

[5 15] 2 setdash 1 setlinecap 3 setlinewidth

265 320 moveto 300 355 340 335 360 320 curveto

stroke

grestore

gsave

[2 8] 2 setdash 2 setlinecap 3 setlinewidth

240 330 moveto 220 340 125 350 140 320 curveto

stroke

grestore

Obrve su također napravljene pomoću Bezierove krivulje, no njihove linije imaju različite atribute.

Određen je setlinecap koji utječe na početni i završni oblik linije. Ta komanda ima jedan parametar vrijednosti koji može biti:

- 0- Kvadratni rub koji završava s definicijom linije
- 1- Zaobljeni rub
- 2- Kvadratni rub, ali produžen za polovicu debljine linije.

Sa setdashom možemo pretvoriti običnu ravnu ispunjenu liniju, u iscrtkanu, točkastu ili kombanciju kraćih i dužih crtica.

U ovom slučaju [5 15] nam kaže da će linija bez obzira na debljinu će crtice biti dužine 5 točaka, a bijelina duljine 15 točaka. 2 setdash određuje to da je prva crtica, ali samo u prvom javljanju kraća za 2 točke. 1 setline cap nam kaže kako će biti zaobljeni rub, kod svake popunjene crtice.

Druga obrva određena je sa [2 8] 2 setdash 2 setlinecap, dakle crtice dužine 2 točke, sa bijelinama od 8 točaka, prva crtica skraćena za dvije točke i korsiti se kvadratni rub koji je produžen za polovicu debljine linije. Obje obrve su definirane sa debljinom od 3 točke ( 3 setlinewidth ), dakle rub je produžen a 1.5 točku.

U Adobe Illustratoru sam pomoću oblika ( rounded rectangle toola i rectangle toola ) nacrtala svaki dio obrve. Pošto su svi osim početnog bili jednake veličine, mijenjala sam im samo smjer rotacije i slagala ih prema krivulji.



Slika 7. Atributi linija

( Izvor: Tara Vagner )

```
%nos
gsave
[0 8] 0 setdash 1 setlinecap 6 setlinewidth
250 270 moveto 230 210 lineto 270 210 lineto
stroke
grestore
gsave
[0 8] 0 setdash 1 setlinecap 2 setlinewidth
255 225 moveto 270 240 270 220 280 218 curveto
stroke
grestore
```

Kod nosa sam koristila najjednostavnije naredbe. Za početni oblik nosa iskoristila sam line moveto i lineto. Kao što je već spomenuto moveto je početna točka, lineto radi liniju, a slijedeći lineto veže se na mjesto na kojem je ta linija završila. [0 8] 0 setdash 1 setlinecap, zapravo nema liniju, ali 1 setlinecap koja određuje zaobljene završetke i zbog toga se napravi krug. Pošto je debljina linije 6 tipografskih točaka, a praznina između tih krugova 8 oni su skoro spojeni.

Drugi dio koji određuje oblik nosa napravljen je sa bezier krivuljom. Ima jednaki setdash i setlinecap kao i prvi dio, ali kod njega je različita debljina linije. Kao što u prvom

dijelu to liniji čini deblju i zbog toga su točkice puno bliže jedna drugoj, u ovom slučaju je linija tanja i točke su udaljene. Točke su udaljene za 6 tipografskih točaka, zato što je praznina određena sa 8, a debljina linije sa 2 tipografske točke.

U Illustratoru kod crtanja nosa upotrijebila sam ellipse tool, te po dužini poslagala krugove jednakih dužina i veličina. U usporedbi sa PostScriptom puno je teže odrediti gdje će se točno nalaziti i koje će veličine biti. Smatram da je u PostScript jeziku puno lakše predvidjeti i izračunati barem otprilike veličinu i izgled, dok u Illustratoru se stvara slika u glavi kako bi to trebalo izgledati.

%lancic

gsave

[0 10] 0 setdash 1 setlinecap 10 setlinewidth

195 50 moveto 305 50 lineto

stroke

grestore

Lancic je napravljen na isti način samo što kod njega debljina linije i praznina je ste numeričke vrijednosti što stvara krugove koji se dodiruju i izgledaju kao da između njih uopće nema razmaka.

Kao i kod nosa u Illustratoru je za lančić korišten ellipse tool, samo što s ovdje svi krugovi bili spojeni i u ravnini, a Illustrator iscertava linije koje određuju gdje se središta nalaze, pa je pomoći njih puno lakše nacrtati nešto u ravnini.

### **3.6. Kružnice i translate**

Krug, kružnica, isječci krugova i njihovi dijelovi definiraju se sa komandama: arc, arcn i arcto. One imaju 5 parametara, a to su koordinate središta, radijus, kut početka i kut završetka. Koordinate središta i radijus se zadaju točkama, a kut početka i završetka u stupnjevima. Naprijer 200 300 100 0 360 arc, 200 je x, 300 y, a 100 veličina radijusa, 0° je kut početka, a 360° kut završetka. Tako bi dobili puni krug u središtu s koordinatama 200 300 i veličine 100 točaka. Kada bi početni kut bio 45°, a završni 0° taj krug bi počinjao od 45°, a završavao na 360°, jer nam 0 kod završnog kuta označava isto što i 360. U arc funkciji

je ugrađen moveto tako da ga ne trebamo ranije koristiti, početak staze se definira sa središtem i radijusom. Arc crta u obrnutom smjeru kazaljke na satu ( do kuta završetka ).

Arcn ( negativan arc ) crta u smjeru kazaljke na satu. [3]

Translate pomiče cijeli koordinatni sustav. U trenutku kada iskoristimo 200 300 translate, tamo nam je zapravo ishodište susatava s točkama  $x=0$  i  $y=0$ .

```
%usta  
  
gsave  
  
1 0 0 setrgbcolor  
  
250 200 translate 0 0 60 220 320 arc  
  
fill  
  
grestore  
  
gsave  
  
1 0 0 setrgbcolor  
  
230 148 translate 0 0 30 150 30 arcn  
  
fill  
  
grestore  
  
gsave  
  
1 0 0 setrgbcolor  
  
270 148 translate 0 0 30 150 30 arcn  
  
fill  
  
grestore
```

Translate je pomakao središte i donju usnu koju sam crtala pomoću naredbe arc. Pošto ona crta u obrnutom smjeru od kazaljke na sat, a željela sam dobiti kružni isječak kojemu je krug prema dolje, odlučila sam se na  $220^\circ$  i  $320^\circ$  zato što su u ravnini i u gornjem dijelu usne se dobije ravna linija. Sa fill se cijeli kružni isječak popunio bojom.

Gornji dio usne sastoji se od dva kružna isječka jednakih veličina, koji su samo pomaknuti po x osi sa naredbom translate. Sa arcn se dobio kružni isječak koji ide u smjeru kazaljke na satu s x i y vrijednostima koje stvaraju ravnu crtu s donje strane, a dio kruga je s gornje strane.

Donji dio usnice u Illustratoru napravljen je pomoću elipse toola. Kada se napravi kružnica , preko nje se stavi rectangle tool na dio kruga koji nam ne treba i gdje se želi dobiti ravna linija. U pathfinderu nalaze se naredbe koje kada su dva oblika preklopljena one ga mogu spojiti, oduzeti oblika jednog od drugog, uzeti samo preklopljeni dio oba oblika ili uzeti samo nepreklopljeni dio oba oblika. U ovom slučaju koristila sam dio koji oduzima oblik jedan od drugoga ( minus front ), dakle kada sam iznad kruga stavila pravokutnik, pravokutnik je obrisao dio kruga koji mi nije bio potreban..



Slika 8. Kružni isjecci

( Izvor: Tara Vagner )

Za gornji dio usnice iskoristila sam donji, njega reflektirala i smanjila do veličine koja mi je bila potrebna za jednu stranu usnice. Pošto je druga strana ista kao i prva, prvu sam kopirala i namjestila da stoji u ravnini s prvom, te u pathfinderu spojila ta dva oblika.

### 3.7. Korištenje repeat petlje

Često u grafikama s likovima dolazi do ponavljanja jednakih linija, koje idu po određenom pravcu, krivudavok putu, kružnici ili simetriji, a kod toga nam služe petlje repeat i for. Komanda repeat počinje planiranim brojem ponavljanja oblika kojeg želimo više puta koristiti. Nakon broja ponavljanja otvara se vitičasta zagrada u koju pišemo izgled i attribute oblika kojeg želimo ponoviti i tek nakon zatvaranje vitičaste zagrade pišemo naredbu repeat. Brojač repeat petlje ne može se iskoristiti kao i varijabla unutar petlje, što for može.

```

%oko lijevo

gsave

/krivulja { 50 10 30 40 10 20 curveto } def

/brojac 40 def

/kut 360 brojac div def

/boja 0 def

190 300 translate

brojac { 1 0.2 0.1 setrgbcolor

2 setlinewidth 0 0 moveto

krivulja stroke kut rotate }repeat

grestore

gsave

190 300 10 0 360 arc

0.7 setgray fill

grestore

```

U ovome dijelu koda korištena je i naredba rotate koja okreće cijeli koordinatni susatav po pozitivno zadanom kutu u stupnjevima u smjeru suprotnom od kazaljke na satu. Kada želimo zarotirati naš oblik, prvo moramo izvršiti rotaciju cijelog koordinatnog sustava, pa tek onda programiranje oblika. [3]

Također ovdje su korištene varijable koje su skup rutina s imenom koje mi odredimo, a pozivaju se kao i bilo koja PostScript komanda.

Div je matematička operacija dijeljenja. Ovdje smo kroz varijablu zadali brojač koji iznosi 40 i drugu proceduru u kojoj kut iznosi 360°. U proceduri /kut 360 brojac div def, kut je ime procedure, 360 stupnjevi za koje će se određeni oblik okretati i početni broj za dijeljenje, brojac označava broj 40 koji je zadan u varijabli prije i označava dijelitelja, div označava matematičku radnju dijeljenja, dakle 360/40, a def definira varijablu. 360 dijeljeno sa 40, daje 9, što označava da će se oblik koji je nacrtan pomicati za svakih 9° po krugu, sve dok ne dođe do 360°.



Unutar naredbe za repeat određena je boja, debljina linije, početna točka, krivulja preko procedure i kut. Unutrašnjost oka napravljena je s krugom, sa središnjim koordinatama istima kao i kod repeat petlje, veličine 10 točaka i sa 30% sivoće.

U Adobe Illustratoru nacrtala sam jednu liniju koja označava krivulju, te ispod nje nacrtala krug sa središtem gdje mi je potreban. Kada se označi ta krivulja i iskoristi rotate tool, sa ctrl i lijevom klikom miša se stiže u središte kružnice te nam izlazi prozor za rotaciju. U prozoru za rotaciju možemo odrediti za koliko će nam se stupnjeva krivulja rotirati po tom središtu. Pritiskom na copy u lijevom uglu tog prozora stvara se krivulja pomknuta za zadani broj stupnjeva, naredbom ctrl+D iscrtavaju se dalje linije po toj osi.

```
%oko desno
```

```
270 300 moveto 280 330 330 340 350 300 curveto
```

```
2 setlinewidth
```

```
Stroke
```

```
270 300 moveto 280 270 330 260 350 300 curveto
```

```
2 setlinewidth
```

```
stroke
```

```
gsave
```

```
/isjecak { Cx 0 r 0 360 arc closepath } def
```

```
/r 9.5 def
```

```
/r2 4 def
```

```
/Cx r r2 add def
```

```
/brojac 50 def
```

```
/kut 360 brojac div def
```

```
310 300 translate
```

```
brojac { 0 0.83 0.60 0 setcmykcolor
```

```
0.2 setlinewidth isjecak stroke
```

```
kut rotate } repeat
```

```
Grestore
```

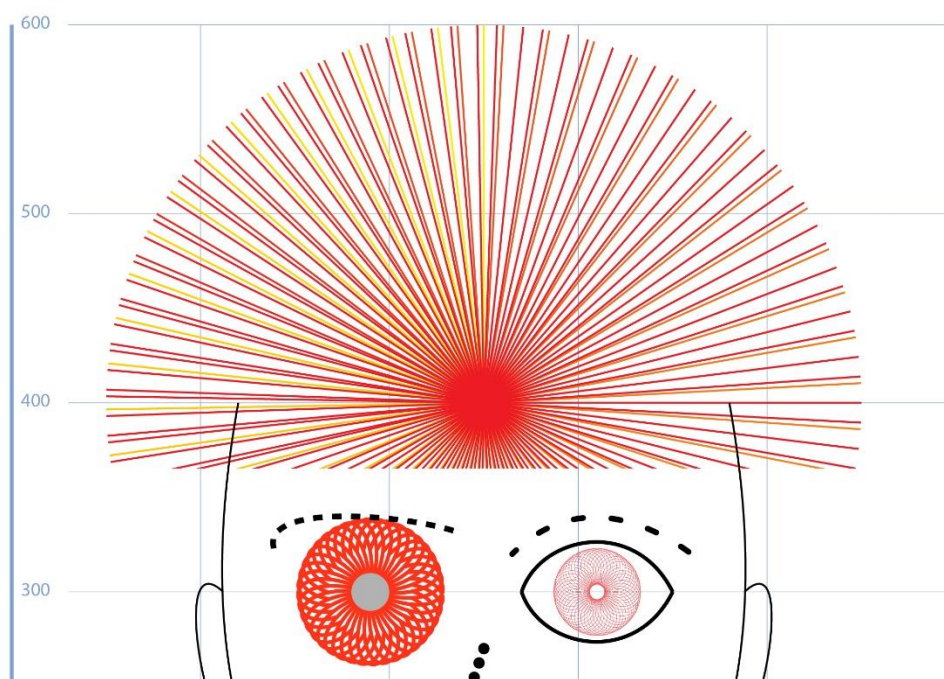
Add je matematička operacija zbrajanja. /Cx r r2 add def, ovdje se zbrajaju dvije varijable r i r2, one iznose 13,5. Iz toga možemo uočiti da je 'isjecak' 13,5 0 9,5 0 360 arc.

Brojač se je odredio kao i u prošloj pomoću naredbe div, ovdje se je svaki put oblik pomakao za 7,2°.

Boja je određena preko setcmykcolor što označava cyan, magenta, yellow i karo, one se također gledaju u postotcima kao i kod setrgbcolor, što u ovom slučaju iznosi 0% cyan, 83% magenta, 60% yellow i 0% karo.

Na isti način kao i kod izrade prvog oka sam izradila i drugo. Pomoću ellipse tool sam nacrtala kružnicu koja je trebala biti ponovljena u krug. Odredila središte luka pomoću manje kružnice na kojem se trebala kretati kružnica, te sa repeat i copy ostvarila željeni rezultat.

Sa pen toolom iscrtala sam dio oko oka, koji je u PostScriptu riješen Bezierovim krivuljama.



Slika 9. Repeat petlja

( Izvor: Tara Vagner )

```
%sesiric
```

```
gsave
```

```
250 400 translate 0 0 200 -10 190 arc
```

```

clip newpath
/b 0 def
/b1 1 def
170 {0 0 moveto 0 200 rlineto
    0 b b1 0 setcmykcolor
    stroke 7 rotate
    /b b 0.016 add def} repeat
Grestore

```

Clip newpath stvara masku od određenog lika, u ovom slučaju od kruga kojeg smo iznad te naredbe stvorili. Za taj krug zadaje se petlja repeat koja se vrti jednu ravnu liniju 170 puta.

Varijable /b 0 def i /b1 1 def su mi u ovo slučaju služile kako bi napravila gradijent boja. B označava boju koja prva kreće, a b1 je završna boja s kojom završava gradijent, u ovom slučaju to se odnosi na prijelaz iz magenete u žutu. Gradijent ne izgleda postepeno jer je s namjerom pomoću 7 rotate napravljeno da sve linije budu izmješane.

Za ovaj dio u Adobe Illustartoru nacrtana je jedna linija i kružnica po kojoj se je ta linija trebala kretat. Ponovno sa repeat, copy i ctrl+D sam sam dobila željeni oblik. No, taj oblik bio je jednobojan, a u PostScriptu je napravljen izmješan gradijent boja, što je u Illustratoru bilo ne moguće. Mijenjajući boje skoro svake linije koja je ponovljena dobio se isti efekt, ali za koji je trebalo jako puno vremena. Dok je u PostScriptu takav složeni dizajn bilo vrlo jednostvano napraviti uz par par varijabli i masku, u Illustratoru to je bilo puno kompliciranije. Donji dio nije se mogao obrisati pomoću path findera i preklapanjem objekta, nego se je morao brisati sa erase toolom, što je inzinstiralo na mirnoj ruci i velikoj preciznosti, dok se u PostScriptu samo odredila veličina maske.

```

%ukrasi
gsave
/nez {0 0 moveto 35 30 40 35 35 40 curveto
    35 30 0 35 0 0 curveto
    40 32 lineto 42 32 lineto

```

```

    closepath} def
300 0 translate 0.5 0.5 scale
    8 {0.4 0.7 0.5 setrgbcolor
    nez fill
    0 20 translate -1 1 scale
    Stroke 45 rotate
    } repeat

```

Grestore

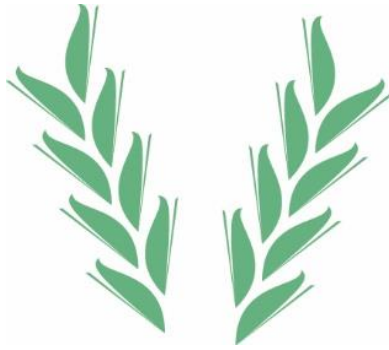
Sa naredbom scale, povećavamo ili smanjujemo naš grafički oblik. Scale iznosi 1, kada je on manji od 1 oblik se smanjuje, a kada je veći od 1 povećava se. Scale ima numeričke vrijednosti x i y. Ako oblik želimo zrcaliti preko nekih od osi onda ispred numeričke vrijednost x ili y stavimo -, ovisno u koju stranu ga želimo zrcaliti.

Pomoću curveto, lineto i closepath naredba napravljen je neobičan oblik sličan listu stavljen u jednu proceduru. Translatirano je središte, a oblik je pomoću naredbe scale smanjen za 50%.

Boja je određena preko setrgb color, a iznosi 40% crvene, 70% zelene i 50% plave. 0 20 translate prikazuje kako se lik pomiće za 20 točaka prema gore ( po y osi ). -1 1 scale zrcali svaki list na jednu stranu. Prvi list dizajniran je tako da gleda na desnu stranu, a pošto je – ispred numeričke vrijednosti x, drugi se zrcali po osi x i onda gleda na lijevu i tako u krug 8 puta, koliko je zadana petlja repeat. Svaki list rotiran je za 45°.

Isti takav kod ponovljen je još jednom kako bi s druge strane bio isti uzorak listova, jedino što je promijenjeno su translate i scale prije repeat varijable. U drugom dijelu koda scale je -0.5 0.5, kako bi se ista ta varijabla ponovila ali zrcaljena po x osi. Također je promijenjen translate i pomaknut 100 točaka bliže y osi.

Pomoću peen toola, nacrtala sam željeni oblik. Tom obliku trebalo je puno točka i popravljjanja zaobljenosti i mjesta tih točaka. Jednu stranu sam samo kopirala i posložila, a za drugu sam morala koristiti zrcaljenje. Boju je bilo dosta teško naći, jer kada se upisuju jednaki postotci koji su korišteni i u PostScriptu, ne ispada ni približno ista.



Slika 10. Bezierova krivulja i repeat

( Izvor: Tara Vagner )

### 3.8. Korištenje for petlje

For petlja kontrolira brojač petlje za koji su date granice i korak napredovanja. Ima 4 parametra: početni brojač, broj koraka, završni brojač i proceduru izvršavanja komande. Početna vrijednost brojača se odmah prikazuje na vrhu stacka. Brojač može imati cjelobrojnu vrijednost, necjelobrojnu, pozitivnu i negativnu, te se on može povećavati ili smanjivati. [3]

```
%nausnice
```

```
113 235 moveto113 150 lineto
```

```
stroke
```

```
387 235 moveto 387 150 lineto
```

```
Stroke
```

Sa lineto su napravljene crne linije koje vode do dizajna naušnica.

```
gsave
```

```
45 -2 4 {/i exch def
```

```
/c i 20 div def
```

```
/c1 1 c sub def
```

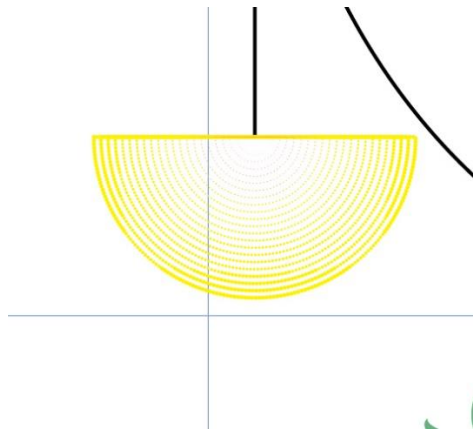
```
0 c1 c 0 setcmykcolor
```

```
[0 1] 0 setdash 1 setlinecap
113 150 i 180 360 arc
i 40 div setlinewidth stroke } for
```

grestore

{/i exch def predstavlja varijablu koja preuzima tekući broj brojača. Tako se kod naredbe za crtanje kružnice gdje je umjesto numeričke vrijednosti za radijus zadan i, kroz cijelu petlju mijenja veličina radijusa. Također kod naredbe za debljinu linija je i 40 div, što označava da se i dijeli sa 40 i tako se mijenja debljina linije.

Sub je matematička operacija oduzimanja, ovdje je zadana u varijabli 1 c sub, gdje se 1 svaki put oduzima sa vrijednosti c, koja je cijelo vrijeme promjenjiva jer je određena preko varijable i 20 div, gdje i preuzima tekući broj brojača i dijeli ga sa 20.



Slika 11. For petlja

( Izvor: Tara Vagner )

U Adobe Illustartoru ovo je zahtjevalo najviše posla. Pokušaj da krug sa gradijentom, preko pixelate-color halftone dobijem identične točkice nije bio moguć, jer kada dođe do dijela kada se treba napraviti image trace, toliko sitne točkice, kolike su meni trebale ispasti, promijenile su svoj oblik. Tada sam odlučila nacrtati toliko kružnih isječaka koliko je i u for petlji, te njih pretvoriti u točkice. Kružnim isječcima sam preko strokea, mijenjala debljinu, završetke, 'kapice' i dash, te dobila slični efekt. Bilo je vrlo komplicirano poslagati kružne isječke kako trebaju stajati, te je svakome se trebala promijeniti boja kako bi se dobio lagani gradijent kao i u PostScriptu.

### 3.9. Učahureni PostScript - eps

Svaki grafički dizajn koji programer napravi u PostScriptu može se koristiti i u Illustratoru. Pomoću BoundingBoxa odredimo koji dio našeg grafičkog dizajna želimo prebaciti u .eps datoteku. Na taj način dobivamo vektorski rezultat našeg rada napravljenog pomoću koda. Također, kada .ps file pretvorimo u .eps u Illustratoru je naš dizajn lako promjenjiv, što je vrlo korisno ako neki dio ne znamo ili ga je teže napraviti u PostScriptu. Naravno može se koristiti i u obrnutim slučajevima kada nam je neki grafički dizajn teže izraditi u Illustratoru, onda se taj dio programira u PostScriptu i pomoću začahurene .eps datoteke prebaci u Illustrator

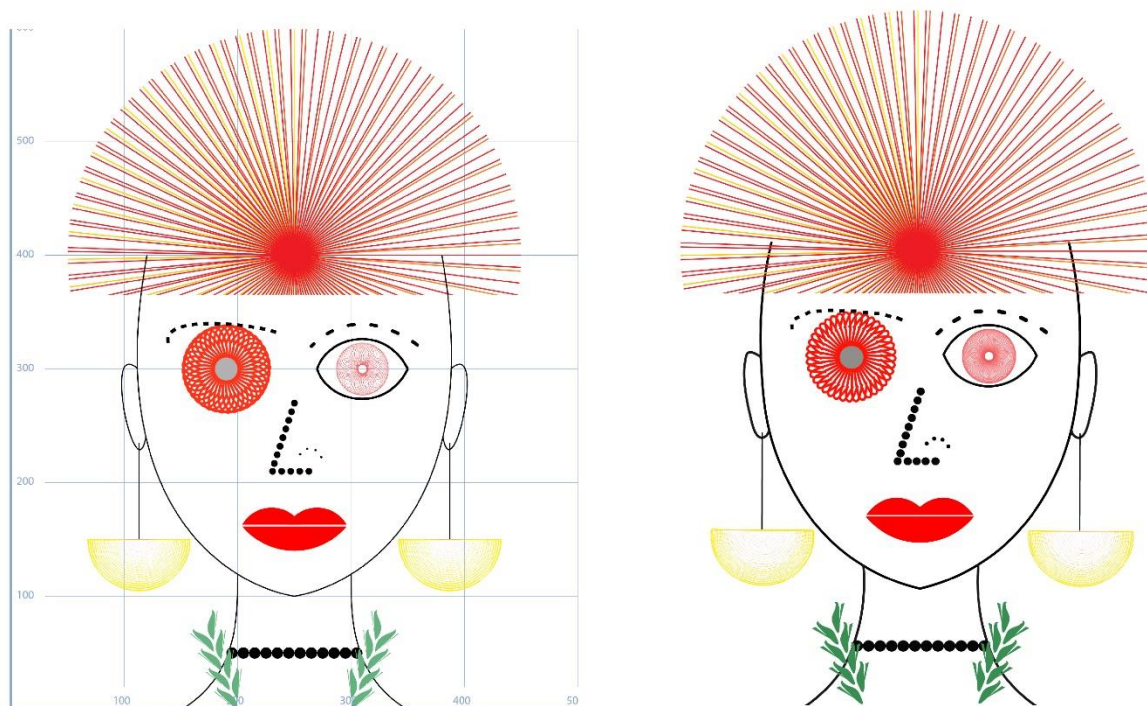
```
%!PS-Adobe-3.0 EPSF-3.0
```

```
%%BoundingBox: 0 100 200 200
```

```
%%EndComments
```

```
%%EndProlog
```

## 4. REZULTATI I RASPRAVA



Slika 12. Usporedba rezultata u PostScriptu i Illustratoru

( Izvor: Tara Vagner )

PostScript i Illustrator pokazali su se kao odlična mjesta za izvršavanje grafičkog dizajna na drugačiji način. Svaki ima svoje prepreke, ali i oba zahtijevaju dobro znanje i snalaženje. PostScript smatram puno bržim, ali možda ne i jednostavnijim. Illustrator, s druge strane ima puno jednostavnih komandi, ali kada dolazi do složenijeg dizajna tu nastaje problem i nije ga toliko jednostavno riješiti. Kod PostScripta vrlo je jednostavno složenije dizajniranje uz dobro znanje varijabla i poznavanje naredbi. Gradijent, ponavljanja i petlje naprave se u nekoliko koraka naredbi s laganim pozicioniranjem i mijenjanjem petlja. Kod Illustratora, sve što se treba ponoviti se svodi na 'copy', iako možemo kopirati po određenom luku kada ga odredimo, puno je teže kasnije ispravljati greške i uglavnom se svodi na to da se radi ispočetka. Vremenski za sve što je u PostScriptu obuhvaćalo repeat petlju mi je trebalo barem 15 minuta manje za izradu, nego u Illustratoru. Za 'naušnice' koje su napravljene preko for petlje, u PostScriptu mi je trebalo otprilike 20 minuta manje, s tim da u Illustratoru ne izgleda odlično kao u PostScriptu.



Kod crtanja zavnutih linija nema previše razlike ni u brzini, ni u jednostavnosti, PostScript za to zahtjeva znanje jednostavnijih naredbi, a Illustrator znanje jednostavnijih alata.

Kod crtanja linija s atributima završetaka i stvaranja bjelina meni je jednostavnije i brže u PostScriptu, jer postavljanjem dasa, debljine linije i završetaka mogu si predočiti otprilike izgled završne linije. Iako se na skoro isti način može napraviti i u Illustratoru 'dashed line' koristeći dash i gap, meni je stvaralo nekoliko problema pri izradi i početnom snalaženju.

## 5. ZAKLJUČAK

Kao što sam i pretpostavila prije uopće započinjanja samoga rada, PostScript je bio puno brži jednostavniji način za izradu grafičkog dizajna. Možda da sam započela i osmislila dizajn u Illustratoru, pa isti takav radila u PostScriptu bi bila obrnuta situacija. No, Illustrator i PostScript, iako su dva potpuno različita načina grafičkog dizajniranja, oba pružaju razne mogućnosti različitog grafičkog dizajna. Smatram da se svatko može naći u jednom od ova dva načina grafičkog dizajniranja. Oba zahtijevaju dosta znanja i poznavanja za korištenje, ali kada se jednom savlada dizajniranje nije problem. Pomoću samih linija mogu se stvoriti vrlo zanimljive grafičke slike, koje izgledaju upotpunjeno.

Kroz poglavlje osnovne naredbe prikazala se jednostavnost i način korištenja najosnovnijih naredbi PostScript jezika, koje su kasnije bile potrebne u svakom dijelu rada. Postavljanje radne površine obuhvaća kompliciranije naredbe, no nije toliko teško za izraditi kada se shvati način korištenja varijabli i petlji, te bez postavljanja radne površine nije moguće dobro snalaženje i nije lako dizajniranje. U poglavlju Bezierove krivulje prikazana je jednostavnost stvaranja linija sa samo četiri točke, od kojih su dvije stabilne i dvije natezne. Memorijski mehanizmi su prikazali održavanje memorije, točnije stanja stacka, prikazano je kako stack uvijek mora biti očišćen prije svakog dijela dizajna ( naredbi/petlji ). U poglavlju atributi za linije se prikazuje mogućnost raznovrsnosti tih atributa, od linija se mogu napraviti točkice, linije sa različitim završecima i sl. Kružnice i translacije prikazuju raznovrsne mogućnosti korištenja krugova, kružnih lukova i kružnih isječaka, a translacije su prikazale mogućnost pomicanja ishodišta, koje nam je kasnije potrebno kod petlji. Korištenjem repeat petlje prikazala se mogućnost dizajna ponavljanjem jednog objekta na različite načine, na vrlo jednostavan način. Dok je korištenje for petlje prikazalo mogućnost i jednostavnost mijenjanja i ponavljanja jednog zadanog objekta na različite načine, taako što mu se mijenjala veličina, debljina, boja, atributi i sl. Izradom učahurenog PostScripta (.eps ) prikazalo se kako se dizajn koji je izrađen preko naredbi, lagano pebacuje u vektorski oblik, te se sa njime može služiti i u Illustratoru.

## 6. LITERATURA

1. Adobe Systems Inc. „PostScript Language Reference Manual”, 1999. godina, Addison-Wesley, Reading, ISBN: 0-201-37922-8
2. <https://whatis.techtarget.com/definition/Postscript>
3. Žiljak, V.; Pap, K. : „Postscript, programiranje grafike“, ISBN: 953-199-000-X, 1998. godina
4. Erin Dill: „The History of Adobe Illustrator“  
<https://www.vecteezy.com/blog/design-tips/the-history-of-adobe-illustrator>
5. IT Connect: „What is Illustrator?“  
<https://itconnect.uw.edu/learn/workshops/online-tutorials/graphics-and-design-workshops/adobe-cs/illustrator/>
6. Altitudetvm: „Što je Adobe Illustrator? Upoznajte definiciju, funkciju i značajke“, s interneta  
<https://altitudetvm.com/hr/komputer/1484-apa-itu-adobe-illustrator-kenali-pengertian-fungsi-dan-fitur-fiturnya.html>
7. Wilson D.; Lourekas P.; Schwartz R. : „ Learn about Adobe Illustrator CC for Graphic Design and Illustration“ , ISBN: 0-13-439778-9, 1996. godina
8. Glenn C. Reid : „Thinking in PostScript“ , ISBN 0-201-52372-8, 1990. godina.

## 7. PRILOZI

```
10 10 translate

/mreza {0.5 0.3 0 0 setcmykcolor

/xOs 500 def

/yOs 600 def

gsave

2 setlinewidth xOs 0 moveto 0 0 lineto 0 yOs lineto stroke

grestore

gsave

0.3 setlinewidth

6 { 30 100 moveto xOs 100 lineto stroke 0 100 translate } repeat

grestore

gsave

0.3 setlinewidth

5 { 100 20 moveto 100 yOs lineto stroke 100 0 translate } repeat

grestore

gsave

/tekst 3 string def /Helvetica findfont 10 scalefont setfont

100 100 yOs { /y exch def 5 y 2 sub moveto y tekst cvs show } for

90 100 xOs { /x exch def x 5 moveto x 10 add tekst cvs show } for

grestore

0 setgray } bind def

mreza

%sesiric

gsave
```

```

250 400 translate 0 0 200 -10 190 arc

clip newpath

/b 0 def

/b1 1 def

170 {0 0 moveto 0 200 rlineto

    0 b b1 0 setcmykcolor  stroke

    7 rotate

    /b b 0.016 add def} repeat

grestore

%oblik lica

120 400 moveto 80 180 190 110 250 100 curveto

stroke

380 400 moveto 420 180 310 110 250 100 curveto

stroke

%vrat

200 120 moveto 200 80 200 30 150 0 curveto

stroke

300 120 moveto 300 80 300 30 350 0 curveto

stroke

%usi

112 300 moveto 80 330 110 190 119 240 curveto

stroke

388 300 moveto 420 330 390 190 381 240 curveto

stroke

%nausnice

113 235 moveto 113 150 lineto

Stroke

```

```

387 235 moveto387 150 lineto

stroke

gsave

45 -2 4 {/i exch def

    /c i 20 div def

    /c1 1 c sub def

        0 c1 c 0 setcmykcolor

        [0 1] 0 setdash 1 setlinecap

        113 150 i 180 360 arc

        113 150 lineto closepath

        i 40 div setlinewidth

        stroke

    } for

grestore

gsave

45 -2 4

    {/i exch def

    /c i 20 div def

    /c1 1 c sub def

        0 c1 c 0 setcmykcolor

        [0 1] 0 setdash 1 setlinecap

        387 150 i 180 360 arc

        387 150 lineto closepath

        i 40 div setlinewidth

        stroke

    } for

grestore

```

```

%oko lijevo

gsave

/krivulja { 50 10 30 40 10 20 curveto } def

/brojac 40 def

/kut 360 brojac div def

/boja 0 def

190 300 translate

brojac { 1 0.2 0.1 setrgbcolor

        2 setlinewidth 0 0 moveto

        krivulja stroke

        stroke kut rotate

    }repeat

grestore

gsave

190 300 10 0 360 arc

0.7 setgray fill

grestore

%oko desno

270 300 moveto 280 330 330 340 350 300 curveto

2 setlinewidth stroke

270 300 moveto 280 270 330 260 350 300 curveto

2 setlinewidth stroke

gsave

/isjecak { Cx 0 r 0 360 arc closepath } def

/r 9.5 def

/r2 4 def

/Cx r r2 add def

```

```

/brojac 50 def

/kut 360 brojac div def

310 300 translate

brojac { 0 r2 r 0 setcmykcolor 0.2 setlinewidth

        isjecak stroke kut rotate

    } repeat

grestore

%obrve

gsave

[5 15] 2 setdash 1 setlinecap 3 setlinewidth

265 320 moveto 300 355 340 335 360 320 curveto

stroke

grestore

gsave

[2 8] 2 setdash 2 setlinecap 3 setlinewidth

240 330 moveto 220 340 125 350 140 320 curveto

stroke

grestore

%nos

gsave

[0 8] 0 setdash 1 setlinecap 6 setlinewidth

250 270 moveto 230 210 lineto 270 210 lineto

stroke

grestore

gsave

[0 8] 0 setdash 1 setlinecap 2 setlinewidth

```



255 225 moveto 270 240 270 220 280 218 curveto

stroke

grestore

%usta

gsave

1 0 0 setrgbcolor

250 200 translate 0 0 60 220 320 arc

fill

grestore

gsave

1 0 0 setrgbcolor

230 148 translate 0 0 30 150 30 arc

fill

grestore

gsave

1 0 0 setrgbcolor

270 148 translate 0 0 30 150 30 arc

fill

grestore

%lancic

gsave

10 setlinewidth [0 10] 0 setdash 1 setlinecap

195 50 moveto 305 50 lineto

stroke

grestore

%ukrasi

gsave

```

/nez { 0 0 moveto 35 30 40 35 35 40 curveto
35 30 0 35 0 0 curveto 40 32 lineto 42 32 lineto
    Closepath } def
300 0 translate
0.5 0.5 scale
    8 { 0.4 0.7 0.5 setrgbcolor nez fill
        0 20 translate -1 1 scale
        Stroke 45 rotate } repeat
grestore
%ukrasi2
gsave
/nez { 0 0 moveto 35 30 40 35 35 40 curveto
35 30 0 35 0 0 curveto
40 32 lineto 42 32 lineto
    Closepath } def
200 0 translate
-0.5 0.5 scale
    8 { 0.4 0.7 0.5 setrgbcolor
        nez fill 0 20 translate
        -1 1 scale stroke
        45 rotate } repeat
grestore

showpage

```